

This Page Is Inserted by IFW Operations  
and is not a part of the Official Record

## **BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning documents *will not* correct images,  
please do not report the images to the  
Image Problem Mailbox.**

**THIS PAGE BLANK (USPTO)**

**GERMAN DEMOCRATIC REPUBLIC**  
**OFFICE FOR INVENTION AND PATENT AFFAIRS**

---

**PATENT SPECIFICATION**

**(12) TRADE PATENT**

Granted in accordance with § 17 paragraph 1 of the Patent Law

**(19) DD (11) 273 910 A1**

**4(51) G 06 F 9/44**

Published in the version submitted by the applicant

---

**(21) WP G 06 F / 317 630 4**

**(22) 06 July 1988**

**(44) 29 November 1989**

---

**(71) VEB Robetron-Buchungsmaschinenwerk Karl-Marx-Stadt, Annaberger Straße 93, Karl-Marx-Stadt, 9010, DD**

**(72) Winkler, Frank, Graduate Engineer; Steifel, Berndt, Doctorate Engineer, DD**

---

**(54) CIRCUIT ARRANGEMENT FOR ELIMINATING UNSTRUCTURED PROGRAM ELEMENTS**

---

**(55) program maintenance, structured programming, unstructured program elements, structural analysis, control structure, Moore-analysis automatic device, Moore-synthesis automatic device, structural transformation**

**(57) The invention finds application in the validation and maintenance of programs written in higher programming languages in computer technology, in particular microcomputer technology. The object of the invention consists in specifying a circuit arrangement that makes possible the inspection of a program written in a higher programming language with respect to the unstructured elements contained in the program and the transforming of these elements such that they conform to the requirements of structured programming. This is achieved in that the control-structure information of a program to be analyzed, which information is output by a microcomputer, is received and analyzed by an analysis circuit, relayed to a synthesis circuit, and fed to an address decoder, which controls a fixed-value memory, from which the semantically-equivalent control-structure information encoded according to structured programming, which information is stored in the memory, reaches the synthesis circuit. Fig. 3**

**ISSN 0433-6461**

**7 pages**

**THIS PAGE BLANK (USPTO)**

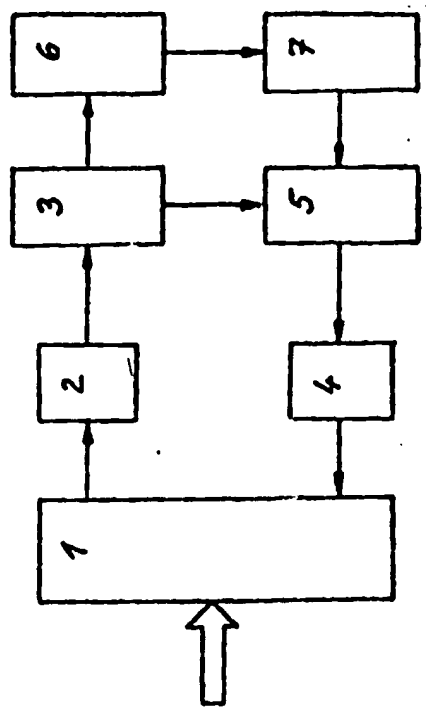
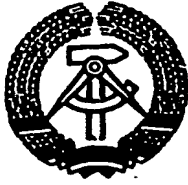


Fig. 3

**THIS PAGE BLANK (USPTO)**



(21) WP G 06 F / 317 630 4

(22) 06.07.88

(44) 29.11.89

(71) VEB Robotron-Buchungsmaschinenwerk Karl-Marx-Stadt, Annaberger Straße 93, Karl-Marx-Stadt, 8010, DD  
(72) Winkler, Frank, Dipl.-Ing.; Stiefel, Berndt, Dr.-Ing., DD

(54) Schaltungsanordnung zur Eliminierung unstrukturierter Programmelemente

(55) Programmwartung, strukturierte Programmierung, unstrukturierte Programmelemente, Strukturanalyse, Steuerstruktur, Moore-Analyseautomat, Moore-Syntheseautomat, Strukturtransformation  
(57) Die Erfindung findet Anwendung bei der Validierung und Wartung von in höheren Programmiersprachen notierte Programmen in der Rechentechnik, insbesondere Mikrorechentechnik. Die Aufgabe der Erfindung besteht darin, eine Schaltungsanordnung anzugeben, die es ermöglicht, ein in einer höheren Programmiersprache notiertes Programm bezüglich der in ihm enthaltenen unstrukturierten Elemente zu untersuchen und diese so zu transformieren, daß sie den Erfordernissen der strukturierten Programmierung entsprechen. Dies gelingt dadurch, daß die von einem Mikrorechner ausgegebenen Steuerstrukturinformationen eines zu analysierenden Programmes von einer Analyseschaltung empfangen und analysiert an eine Syntheseschaltung übergeben sowie einem Adreßdecoder zugeführt werden, welcher einen Festwertspeicher ansteuert, von dem die dort abgelegten, semantisch äquivalenten, der strukturierten Programmierung entsprechenden codierten Steuerstrukturinformationen zur Syntheseschaltung gelangen. Fig. 3

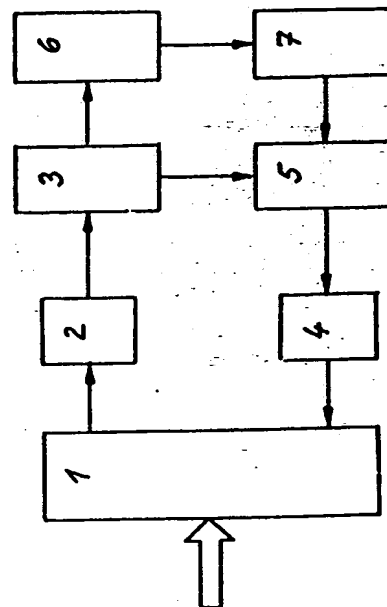


Fig. 3

### Patentansprüche:

1. Schaltungsanordnung zur Eliminierung unstrukturierter Programmelemente, dadurch gekennzeichnet, daß eine Analyseschaltung (3.1.) einerseits mit einer Syntheseschaltung (3.4.) und andererseits über einen Adreßdecoder (3.3.) mit einem Festwertspeicher (3.2.) verbunden ist, dessen Ausgänge ebenfalls an die Syntheseschaltung (3.4.) geführt sind in der Arbeitsweise, daß die Steuerstrukturinformationen des zu analysierenden Programmes von der Analyseschaltung empfangen und analysiert an die Syntheseschaltung (3.4.) und den Adreßdecoder (3.3.) übergeben werden, damit die der unstrukturierten Steuerstruktur entsprechenden, in dem Festwertspeicher abgelegten codierten Steuerstrukturinformationen zur Syntheseschaltung (3.4.) gelangen.
2. Schaltungsanordnung nach Anspruch 1, dadurch gekennzeichnet, daß die Syntheseschaltung (3.4.) die von der Analyseschaltung (3.1.) gelieferten unstrukturierten Steuerstrukturinformationen durch semantisch äquivalente, wohlstrukturierte Steuerstrukturinformationen, die in dem Festwertspeicher (3.2.) abgelegt sind, ersetzt.
3. Schaltungsanordnung nach Anspruch 1 und 2, dadurch gekennzeichnet, daß Analyseschaltung (3.1.) und Syntheseschaltung (3.4.) als Moore-Automaten ausgebildet sind.
4. Schaltungsanordnung nach Anspruch 1 bis 3, dadurch gekennzeichnet, daß die Eingänge der Analyseschaltung (3.1.) und die Ausgänge der Syntheseschaltung (3.4.) über Ports mit einem Mikrorechner verbunden sind, der das zu analysierende Programm an die Analyseschaltung (3.1.) ausgibt und das transformierte Programm von der Syntheseschaltung (3.4.) empfängt.

Hierzu 3 Seiten Zeichnungen

### Anwendungsgebiet

Die Erfindung findet Anwendung bei der Validierung und Wartung von in höheren Programmiersprachen notierten Programmen in der Rechentechnik, insbesondere Mikrorechentechnik.

### Stand der Technik

Die durchgehende Strukturierung von Programmen ist in der Rechentechnik, insbesondere Mikrorechentechnik, von großem Nutzen, da auf dieser Basis Verifikation, Validierung und Wartung von Programmen beschleunigt und wesentlich erleichtert werden. Aus einer Reihe von Veröffentlichungen ist es daher bereits bekannt, bei der deshalb notwendigen Transformation von unstrukturierten Programmen in strukturierte alle Sprungbefehle zu substituieren. Das führt dazu, daß gekünstelte Konstruktionen entstehen, welche die Verständlichkeit des Programmes und damit seine Handhabungseffektivität bei Validierung und Wartung verringern. Das ist bedingt durch den Einsatz solcher Mittel wie Codecopierung, Generierung von parameterlosen Unterprogrammen, Einführung zusätzlicher Steuervariablen. Als Beispiele für diesen Lösungsweg seien genannt:

- Ashcroft E., Manna Z.: Translating program schemas to while schemas. S/AM J. 2 (1975), pp 125-148;
- Knuth D.E., Floyd R.W.: Notes and avoiding „go to“ statements. Inför. Proc Letters 1 (1971), pp 23-31;
- Peterson W.W., Kasami, T., Tokura N.: On the capabilities of while, repeat and exit statements. Comm.ACM 16, 8 (Aug. 1973), pp 503-512.

Einen Fortschritt stellt bereits die Lösung von Baker B.S.: An Algorithm for Structuring Flowgraphs; J.ACM Vol. 24, No. 1 (Jan. 1977), pp 98-120 dar, der die Verwendung von „go to“ in beschränktem Maße bei irreduziblen Flußgraphen zuläßt.

Allerdings weist diese Lösung noch zwei wesentliche Mängel auf:

- Ausgangspunkt der Transformation ist der Flußgraph des Programms,
- es besteht Abhängigkeit von der benutzten Programmiersprache.

In Fällen, in denen die benutzte Programmiersprache die strukturierte Programmierung weder unterstützt noch erzwingt, ist es darüber hinaus bekannt, Preprozessoren einzusetzen. Diese führen eine Modellierung von zusätzlichen, die strukturierte Programmierung unterstützenden Operatoren durch die der Sprache eigenen Operatoren durch (z.B. Kernighan B.W., Plauger P.L.: Programmierwerkzeuge; Springer-Verlag Berlin, Heidelberg, New York 1980). Die Verwendung von Preprozessoren ist äquivalent der Realisierung einer neuen Sprache (z.B. Ratfor). Damit wird aber nicht vorhandene Software berührt, deren Handhabungseffektivität vor allem in der Wartungsphase erhöht werden soll.

### Ziel der Erfindung

Das Ziel der Erfindung besteht darin, eine hohe Flexibilität bei der Eliminierung unstrukturierter Programmelemente zu erreichen und dabei schwer verständliche, künstlich-sterile Strukturen zu vermeiden.



## Aufgabe der Erfindung

Der Erfindung liegt die Aufgabe zugrunde, eine Schaltungsanordnung zur Eliminierung unstrukturierter Programmelemente anzugeben, mit der es möglich ist, nichtstrukturierte Programmelemente von in höheren Programmiersprachen notierten Programmen zu erkennen und selektiv so zu transformieren, daß sie den Regeln der wohlstrukturierten Programmierung entsprechen.

Erfindungsgemäß gelingt das dadurch, daß eine Analyseschaltung einerseits mit einer Syntheschaltung und andererseits über einen Adreßdecoder mit einem Festwertspeicher verbunden ist, dessen Ausgänge ebenfalls an die Syntheschaltung geführt sind, wobei die Steuerstrukturinformationen des zu analysierenden Programmes von der Analyseschaltung empfangen und analysiert an die Syntheschaltung und den Adreßdecoder übergeben werden, damit die der unstrukturierten Steuerstruktur entsprechenden, in dem Festwertspeicher abgelegten codierten Steuerstrukturinformationen zur Syntheschaltung gelangen. Die Erfindung umfaßt ferner, daß die Syntheschaltung die von der Analyseschaltung gelieferten unstrukturierten Steuerstrukturinformationen durch semantisch äquivalente, wohlstrukturierte Steuerstrukturinformationen, die in dem Festwertspeicher abgelegt sind, ersetzt. Vorzugsweise können dabei Analyseschaltung und Syntheschaltung als Moore-Automaten ausgebildet sein. Ein Merkmal der Erfindung ist schließlich, daß die Eingänge der Analyseschaltung und die Ausgänge der Syntheschaltung über Ports mit einem Mikrorechner verbunden sind, der das zu analysierende Programm an die Analyseschaltung ausgibt und das transformierte Programm von der Syntheschaltung empfängt.

## Ausführungsbeispiel

Die Erfindung wird im folgenden anhand von Zeichnungen erläutert. Diese zeigen:

Fig. 1: einen Struktographen eines ursprünglichen, nicht den Prinzipien der strukturierten Programmierung entsprechenden Programmsegmentes;

Fig. 2: den Struktographen des transformierten Programmsegmentes von Fig. 1;

Fig. 3: das Blockschema einer Schaltungsanordnung zur Eliminierung unstrukturierter Programmelemente.

Der in Fig. 1 wiedergegebene, der Programmstruktur äquivalente Struktograph weist mit dem Sprungbefehl „GOTO M1“ ein nichtstrukturiertes Programmelement auf, welches durch Transformation (Codekopierung) so zu substituieren ist, daß ein gleichwirkendes Programmsegment entsteht, welches nur noch der wohlstrukturierten Programmierung entsprechende Programmelemente enthält (Fig. 2). Dies geschieht dadurch, daß an die Stelle des unerwünschten Sprungbefehls die nachfolgend zutreffende, semantisch äquivalente allgemeine Anweisung „AA3“ tritt.

Zur Durchführung dieser Maßnahmen dient eine Schaltungsanordnung, wie sie in Fig. 3 beispielhaft gezeigt ist. Darin inbegriffen ist ein Mikrorechner 1, der über einen I/O-Port 2 mit einer Analyseschaltung 3 und über einen I/O-Port 4 mit einer Syntheschaltung 5 verbunden ist. Die Analyseschaltung 3, im Ausführungsbeispiel ein Moore-Analyseautomat, besitzt ausgangsseitige Verbindungen, zu der bereits oben erwähnten Syntheschaltung 5, die in analoger Weise als Moore-Syntheseautomat ausgebildet ist, und über einen Adreßdecoder 6 zu einem Festwertspeicher 7, dessen Ausgang schließlich ebenfalls an die Syntheschaltung 5 führt. Sowohl Moore-Analyse- als auch -syntheseautomat werden zweckmäßigerweise als programmierbare Logikanordnungen (PLA) realisiert.

### Wirkungsweise

Das in einer höheren Programmiersprache vorliegende, zu überprüfende und im Bedarfsfall zu transformierende Programm wird nach seiner Eingabe in den Mikrorechner 1 mit dessen Hilfe einer Strukturanalyse unterzogen und dabei mittels dynamischer Variabler in eine verkettete Liste überführt, die eine direkte Darstellungsform der Struktur des analysierten Programms ist.

Über den I/O-Port 2 erfolgt die Ausgabe dieser verketteten Liste, d.h. die Übergabe der diesbezüglichen Steuerstrukturinformationen an die Analyseschaltung 3. Beim Auftreten eines in der Struktur der verketteten Liste sich widerspiegelnden, nicht den Regeln der wohlstrukturierten Programmierung entsprechenden Programmabschnitts bzw. -elements geht die Analyseschaltung 3 in einen diesen „Fehler“ identifizierenden Zustand über und gibt eine codierte Adresse zum Adreßdecoder 6, der damit den als EPROM realisierten Festwertspeicher 7 ansteuert. Von dessen Speicherplätzen wird im Ergebnis ein aus semantisch äquivalenten, wohlstrukturierter Steuerstrukturinformationen bestehender Befehlssatz gelesen, der in der Weise zur Steuerung der Syntheschaltung 5 dient, daß diese in der Lage ist, eine Umstrukturierung des betreffenden Teiles der ihr von der Analyseschaltung übergebenen verketteten Liste vorzunehmen. Diese korrigierte, nunmehr den Erfordernissen der strukturierten Programmierung entsprechende verkettete Liste gelangt über den I/O-Port 4 zum Mikrorechner 1 zurück und wird von diesem in die höhere Programmiersprache, in der das Programm zu Beginn notiert war, rücktransformiert.

Sowohl die Ausgabe der ursprünglichen verketteten Liste vom Mikrorechner 1 an die Analyseschaltung 3 als auch die Rückgabe der korrigierten verketteten Liste von der Syntheschaltung 5 an den Mikrorechner 1 erfolgen im Dialogbetrieb, also im Wechsel zwischen Listner- und Talkermodus.

Werden von der Analyseschaltung 3 keine nichtstrukturierten Programmelemente festgestellt, arbeitet auch die Syntheschaltung 5 transparent und gibt die ihr zugeführten Informationen über den I/O-Port 4 unverändert wieder in den Mikrorechner 1 aus.

Gegenüber den bekannten Verfahren, insbesondere dem nachträglichen Strukturieren von Programmen „per Hand“ weist die Schaltungsanordnung eine Reihe wesentlicher Vorteile auf. So ist unter Verwendung eines normalen I/O-Ports die Kopplung an ein Mikrorechnersystem problemlos möglich und damit Voraussetzung gegeben, den angeschlossenen Mikrorechner sowohl für die Konstruktion der verketteten Liste als auch für deren Rücküberführung in die ursprüngliche Darstellung form abwechselnd und damit äußerst effektiv zu nutzen. Aufgrund des einfachen, unorthodoxen Konzepts können Programme in sehr kurzer Zeit auf unstrukturierte Elemente überprüft und diese eliminiert werden, da keine aufwendigen Rechenoperationen durchgeführt werden müssen. Die Schaltungsanordnung zur Eliminierung unstrukturierter Programmelemente ist an keine spezielle höhere Programmiersprache gebunden, sondern universell einsetzbar. Der gesamte Analyse- und Eliminierungsprozeß geschieht automatisch, so daß außer dem Starten des zu bearbeitenden Programmes und der Abnahme des Bearbeitungsergebnisses keine weiteren Aktivitäten erforderlich sind.

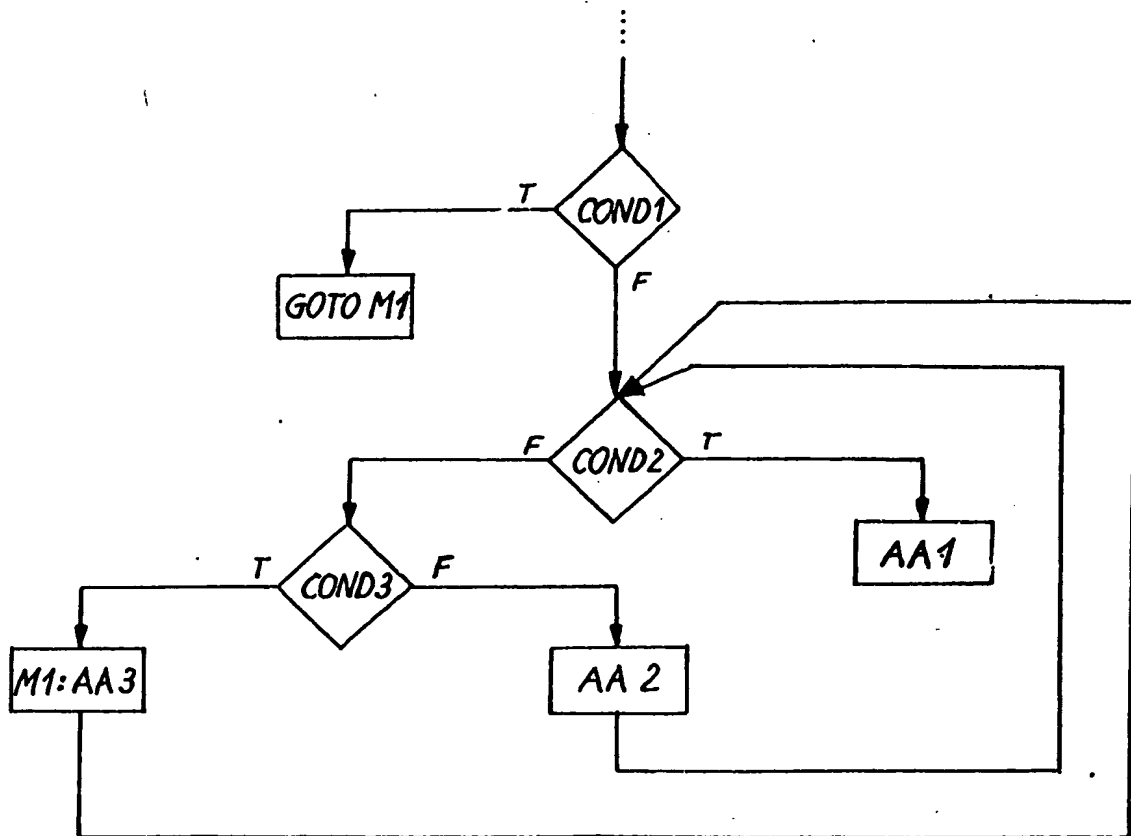


Fig. 1

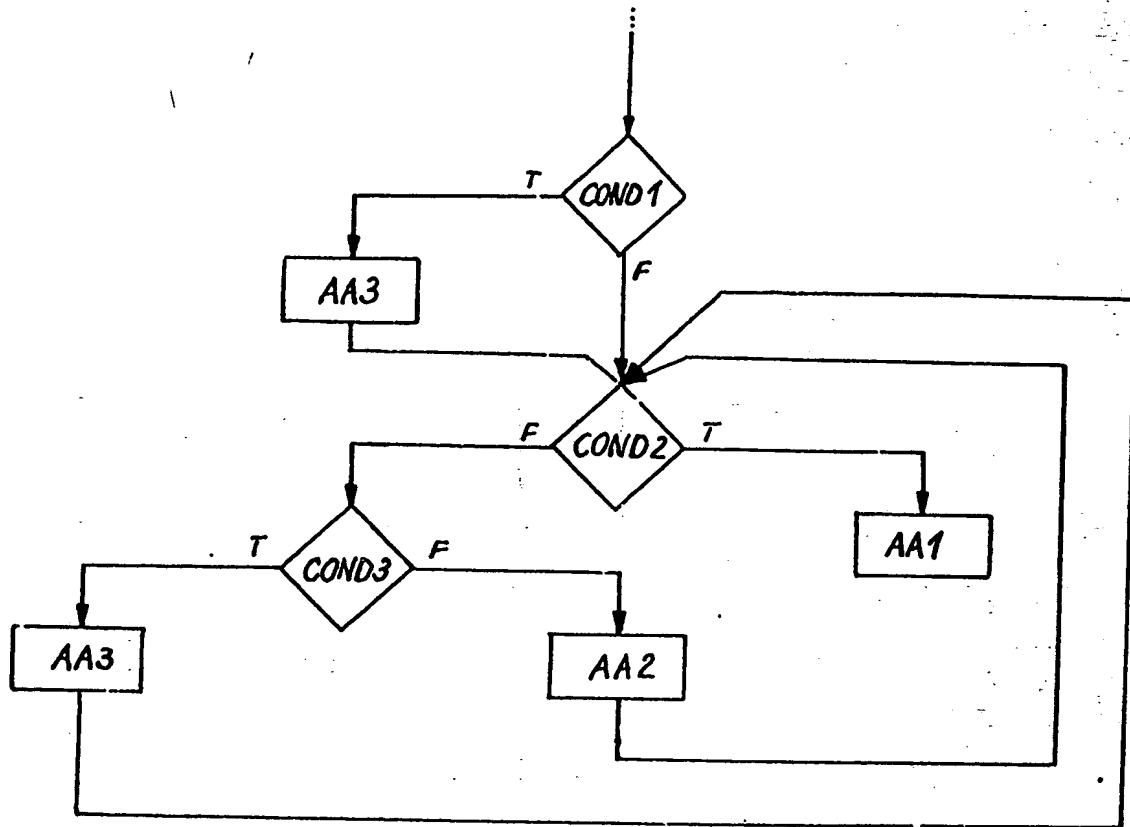


Fig. 2

273910

-6-

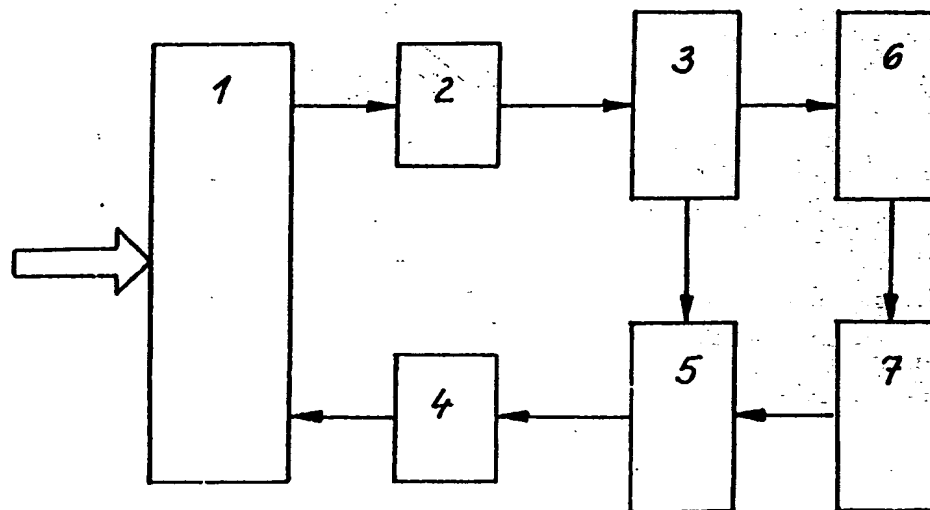


Fig. 3

**THIS PAGE BLANK (USPTO)**